Chapter 7

Configure Protocol Family and Address Interface Properties

For each logical interface, you must configure one or more protocol families and you can configure interface address properties. To do this, you can include the following statements at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]
family family {
    accounting {
        destination-class-usage;
        source-class-usage {
            (input | output | [input output]);
        }
    }
    bundle ml-fpc/pic/port;
    filter {
        input filter-name;
        output filter-name;
        group filter-group-number;
    }
    ipsec-sa sa-name;
    mtu bytes;
    multicasts-only;
    no-redirects;
    policer {
        input policer-template-name;
        output policer-template-name;
    }
    primary;
    remote {
        mac-address address;
    }
    rpf-check fail-filter filter-name;
    address address {
        arp ip-address (mac | multicast-mac) mac-address <publish>;
        destination destination-address;
        eui-64;
        broadcast address;
        multipoint-destination destination-address (dlci dlci-identifier | vci vci-identifier);
```

```
            multipoint-destination destination-address {
                inverse-arp;
                oam-liveness {
                    up-count cells;
                    down-count cells;
                }
                oam-period seconds;
                shaping {
                    (cbr rate | vbr peak rate sustained rate burst length);
                    queue-length number;
                }
                vci vpi-identifier.vci-identifier;
            }
            preferred;
            primary;
            vrrp-group group-number {
                virtual-address [addresses];
                priority number;
                (accept-data | no-accept-data);
                advertise-interval seconds;
                authentication-type authentication;
                authentication-key key;
                (preempt | no-preempt);
                track {
                    interface interface-name priority-cost cost;
                }
            }
        }
    }
```

This chapter describes the interface protocol and address properties that you can configure:

Enable Source Class and Destination Class Usage on page 147

Configure Multilink Interfaces on page 151

Configure Security Associations on page 153

Configure VRRP on page 154

Configure the Protocol Family

For each logical interface, you can configure one or more of the following protocols that run on the interface:

ccc—Circuit Cross-Connect (CCC). You can configure this protocol family for the logical interface of CCC physical interfaces. When you use this encapsulation, you can configure the family ccc only.

inet—IP (Internet Protocol). You must configure this protocol family for the logical interface to support IP protocol traffic, including OSPF, BGP, and ICMP.

inet6—IP (Internet Protocol) version 6. You must configure this protocol family for the logical interface to support IPv6 protocol traffic, including RIPng, IS-IS, and BGP. For more information about IPv6, see “IPv6 Introduction” on page 128.

iso—ISO. You must configure this protocol family for the logical interface to support IS-IS traffic.

mlfr—Multilink Frame Relay. You must configure this protocol (or MLPPP) for the logical interface to support multilink bundling.

multilink-ppp—Multilink Point-to-Point Protocol (MLPPP). You must configure this protocol (or MLFR) for the logical interface to support multilink bundling.

mpls—Multiprotocol Label Switching (MPLS). You must configure this protocol family for the logical interface to participate in an MPLS path.

tcc—Translational Cross-Connect (TCC). You can configure this protocol family for the logical interface of TCC physical interfaces. When you use this encapsulation, you can configure the family tcc only.

tnp—Trivial Network Protocol. This protocol is used to communicate between the Routing Engine and the System Control Board (SCB), System and Switch Board (SSB), Forwarding Engine Board (FEB), or System and Forwarding Module (SFM), depending on router model, in the router’s Packet Forwarding Engine. The JUNOS software automatically configures this protocol family on the router’s internal interfaces only, as discussed in “Configure the Internal Ethernet Interface” on page 247.

To configure the logical interface's protocol family, include the family statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level, specifying the selected family. To configure more than one protocol family on a logical interface, include multiple family statements. Following is the minimum configuration:

```
[edit interfaces interface-name unit logical-unit-number]
family family {
    mtu size;
    multicasts-only;
    no-redirects;
    primary;
    address address {
        destination address;
        broadcast address;
        preferred;
        primary;
    }
}
```

IPv6 Introduction

IPv4 has been widely deployed and used to network the Internet today. With the rapid growth of the Internet, enhancements to IPv4 are needed to support the influx of new subscribers, Internet-enabled devices, and applications. IPv6 is designed to enable the global expansion of the Internet.

IPv6 builds upon the functionality of IPv4, providing improvements to addressing, configuration and maintenance, and security.

IPv6 is defined in the following documents:

RFC 2460, *Internet Protocol, Version 6 (IPv6)*

RFC 2373, *IP Version 6 Addressing Architecture*

IPv4-to-IPv6 Transition

Implementing IPv6 requires a transition mechanism to allow interoperability between IPv6 nodes (both routers and hosts) and IPv4 nodes. The transition mechanism is the key factor in the successful deployment of IPv6. Because millions of IPv4 nodes already exist, upgrading every node to IPv6 at the same time is not feasible.

As a result, transition from IPv4 to IPv6 happens gradually, allowing nodes to be upgraded independently and without disruption to other nodes. While a gradual upgrade occurs, compatibility between IPv6 and IPv4 nodes becomes a requirement. Otherwise, an IPv6 node would not be able to communicate with an IPv4 node.

Transition mechanisms allow IPv6 and IPv4 nodes to coexist together in the same network, and make gradual upgrading possible. The transition mechanism supported by the JUNOS Internet software is tunneling. Tunnels allow IPv6 packets to be encapsulated into IPv4 headers and sent across an IPv4 infrastructure. For more information about configuring tunnels to support IPv4-to-IPv6 transition, see "Configure an IPv6 over IPv4 Tunnel" on page 315.

Configure the Interface Address

You assign an address to an interface by specifying the address when configuring the protocol family. For the inet family, you configure the interface's IP address. For the iso family, you configure an address for the loopback interface. For the ccc, tcc, mpls, and tnp families, you never configure an address.

The JUNOS software also supports IS-IS addresses on interfaces other than lo0, such as T1, T3, Fast Ethernet, SONET/SDH, and ATM interfaces. This can be useful when you are running multiple instances of IS-IS. For more information about running multiple instances of IS-IS, see the *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*.

To assign an address to an interface, include the address statement at the [edit interfaces *interface-name* unit *logical-unit-number* family *family*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family family]
address address {
    destination address;
    eui-64;
    broadcast address;
    preferred;
    primary;
}
```

In the address statement, specify the network address of the interface.

For each address, you can optionally configure one or more of the following:

Address of the remote side of the connection (for point-to-point interfaces only)—Specify this in the destination statement.

Whether the router automatically generates the host number portion of interface addresses—The eui-64 statement applies only to interfaces that carry IPv6 traffic, where the prefix length of the address is 64 bits or less, and the low-order 64 bits of the address are zero. This option does not apply to the loopback interface (lo0) because IPv6 addresses configured on the loopback interface must have a 128-bit prefix length.

Broadcast address for the interface's subnet—Specify this in the broadcast statement; this applies only to Ethernet interfaces, such as the management interface fxp0, the Fast Ethernet interface, and the Gigabit Ethernet interface.

Whether this address is the preferred address—Each subnet on an interface has a preferred local address. If you configure more than one address on the same subnet, the preferred local address is chosen by default as the source address when you originate packets to destinations on the subnet. For more information about preferred addresses, see "Configure Default, Primary, and Preferred Addresses and Interfaces" on page 132.

By default, the preferred address is the lowest numbered address on the subnet. To override the default and explicitly configure the preferred address, include the preferred statement when configuring the address.

Whether this address is the primary address—Each interface has a primary local address. If an interface has more than one address, the primary local address is used by default as the source address when you originate packets out the interface where the destination gives no hint about the subnet (for example, some ping commands). For more information about primary addresses, see "Configure Default, Primary, and Preferred Addresses and Interfaces" on page 132.

By default, the primary address on an interface is the lowest numbered non-127 preferred address on the interface. To override the default and explicitly configure the preferred address, include the primary statement when configuring the address.

Configure the IPv6 Address on an Interface

You represent IPv6 addresses in hexadecimal notation using a colon-separated list of 16-bit values.

You assign a 128-bit IPv6 address to an interface by including the address statement at the [edit interfaces *interface-name* unit *logical-unit-number* family inet6] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family inet6]
address aaaa:bbbb:...:zzzz/nn;
```

The double colon (::) represents all bits set to 0, as shown in the following example:

```
interfaces fe-0/0/1 {
    unit 0 {
        family inet6 {
            address fec0:1:1:1::2/64;
        }
    }
}
```

Configure an Unnumbered Interface

When you need to conserve IP addresses, you can configure unnumbered interfaces. To do this, configure the protocol family, but do not include the address statement at the [edit interfaces *interface-name* unit *logical-unit-number* family *family*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]
family family;
```

For example:

```
[edit]
interfaces {
    so-6/1/0 {
        unit 0 {
            family inet;
            family iso;
        }
    }
}
```

When configuring unnumbered interfaces, you must ensure that a source address is configured on some interface in the router. This address is the default address. We recommend that you do this by assigning an address to the loopback interface (lo0), as described in “Configure the Loopback Interface” on page 259. If you configure an address (other than a martian) on the lo0 interface, that address is always the default address, which is preferable because the loopback interface is independent of any physical interfaces and therefore is always accessible.

Set the Protocol MTU

For each interface, you can configure an interface-specific MTU by including the mtu statement at the [edit interfaces *interface-name*] hierarchy level. If you need to modify this MTU for a particular protocol family, include the mtu statement at the [edit interfaces *interface-name* unit *logical-unit-number* family *family*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family family]
mtu bytes;
```

The default protocol MTU is 4470 bytes for ATM PVC, Cisco HDLC, Frame Relay, and PPP encapsulations. For Ethernet encapsulation on IPv4, the default protocol MTU is 1500 bytes. For Ethernet encapsulation on ISO, the default protocol MTU is 1497 bytes.

When you initially configure an interface, the protocol MTU is calculated automatically. However, if you subsequently change the media MTU, the protocol MTU on existing address families does not automatically adjust.

If you increase the size of the protocol MTU, you must ensure that the size of the media MTU is equal to or greater than the sum of the protocol MTU and the encapsulation overhead. If you reduce the media MTU size, but there are already one or more address families configured and active on the interface, you must also reduce the protocol MTU size. (You configure the media MTU by including the mtu statement at the [edit interfaces *interface-name*] hierarchy level, as discussed in “Configure the Media MTU” on page 41.)

For Ethernet encapsulation when the family is mpls, the default protocol MTU is 1500 bytes, including 4 to 12 bytes of overhead. The maximum number of DCLIs is determined by the MTU on the interface. If you have keepalives enabled, the maximum number of DLCIs is 1000, with the MTU set to 5012.

The actual frames transmitted also contain cyclic redundancy check (CRC) bits, which are not part of the MTU. For example, the default protocol MTU for a gigabit Ethernet interface is specified as 1500 bytes, but the largest possible frame size is actually 1504 bytes; you need to consider the extra bits in calculations of MTUs for interoperability.

Disable the Sending of Redirect Messages on an Interface

By default, the interface sends protocol redirect messages. To disable the sending of these messages on an interface, include the no-redirects statement at the [edit interfaces *interface-name* unit *logical-unit-number* family *family*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family family]
no-redirects;
```

To disable the sending of protocol redirect messages for the entire router, include the no-redirects statement at the [edit system] hierarchy level.

Configure Default, Primary, and Preferred Addresses and Interfaces

The router has a default address and a primary interface, and interfaces have primary and preferred addresses.

The *default address* of the router is used as the source address on unnumbered interfaces. The routing protocol process tries to pick the default address as the router ID, which is used by protocols, including OSPF and IBGP.

The *primary interf ace* for the router is the interface that packets go out when no interface name is specified and when the destination address does not imply a particular outgoing interface.

An interface's *primary addr ess* is used by default as the local address for broadcast and multicast packets sourced locally and sent out the interface. An interface's *preferred address* is the default local address used for packets sourced by the local router to destinations on the subnet.

The default address of the router is chosen using the following sequence:

1. The primary address on the loopback interface lo0 that is not 127.0.0.1 is used.
2. The primary address on the primary interface is used.

To configure these addresses and interfaces, you can do the following:

Configure the Primary Interface for the Router on page 133

Configure the Primary Address for an Interface on page 133

Configure the Preferred Address for an Interface on page 133

Configure the Primary Interface for the Router

The *primary interface* for the router has the following characteristics:

- It is the interface that packets go out when you type a command such as ping 255.255.255.255—that is, a command that does not include an interface name (there is no interface *type*-0/0/0.0 qualifier) and where the destination address does not imply any particular outgoing interface.
- It is the interface on which multicast applications running locally on the router, such as SAP, do group joins by default.
- It is the interface from which the default local address is derived for packets sourced out an unnumbered interface if there are no non-127 addresses configured on the loopback interface, lo0.

By default, the multicast-capable interface with the lowest-index address is chosen as the primary interface. If there is no such interface, the point-to-point interface with the lowest index address is chosen. Otherwise, any interface with an address could be picked. In practice, this means that, on the router, the fxp0 interface is picked by default.

To configure a different interface to be the primary interface, include the primary statement at the [edit interfaces *interface-name* unit *logical-unit-number* family *family*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family family]
primary;
```

Configure the Primary Address for an Interface

The *primary address* on an interface is the address that is used by default as the local address for broadcast and multicast packets sourced locally and sent out the interface. For example, the local address in the packets sent by a ping interface so-0/0/0.0 255.255.255.255 command is the primary address on interface so-0/0/0.0. The primary address flag also can be useful for selecting the local address used for packets sent out unnumbered interfaces when multiple non-127 addresses are configured on the loopback interface, lo0. By default, the primary address on an interface is selected as the numerically lowest local address configured on the interface.

To set a different primary address, include the primary statement at the [edit interfaces *interface-name* unit *logical-unit-number* family *family* address *address*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family family address address]
primary;
```

Configure the Preferred Address for an Interface

The *preferred address* on an interface is the default local address used for packets sourced by the local router to destinations on the subnet. By default, the numerically lowest local address is chosen. For example, if the addresses 128.100.1.1/24, 128.100.1.2/24, and 128.100.1.3/24 are configured on the same interface, the preferred address on the subnet (by default, 128.100.1.1) would be used as a local address when you issue a ping 128.100.1.5 command.

To set a different preferred address for the subnet, include the preferred statement at the [edit interfaces *interface-name* unit *logical-unit-number* family *family* address *address*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family family address address]
preferred;
```

Configure a Point-to-Multipoint ATM Connection

To configure a point-to-multipoint (NBMA) ATM connection, include the multipoint-destination statement at the [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family inet address address]
multipoint-destination destination-address vci vpi-identifier.vci-identifier;
```

You can also include the multipoint-destination statement at the [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family inet address address]
multipoint-destination destination-address {
    inverse-arp;
    oam-liveness {
        up-count cells;
        down-count cells;
    }
    oam-period seconds;
    shaping {
        (cbr rate | vbr peak rate sustained rate burst length);
        queue-length number;
    }
    vci vpi-identifier.vci-identifier;
}
```

address is the interface's address. The address must include the destination prefix (for example, /24).

For each destination, include one multipoint-destination statement. *destination-address* is the address of the remote side of the connection, and *vci-identifier* and *vpi-identifier* are the VCI and optional VPI identifiers for the connection.

When you are configuring point-to-multipoint connections, all interfaces in the subnet must use the same MTU size.

When configuring point-to-multipoint ATM connections, you can do the following:

Configure Inverse ATM ARP on page 135

Configure Inverse Frame Relay ARP on page 135

Define the ATM Traffic-Shaping Profile on page 135

Define the ATM OAM F5 Loopback Cell Period on page 139

Configure the ATM OAM F5 Loopback Cell Threshold on page 140

Configure Inverse ATM ARP

You can configure ATM interfaces to support inverse ATM ARP, as described in RFC 2225. When inverse ATM ARP is enabled, the router responds to received inverse ATM ARP requests by providing IP address information to the requesting ATM device.

The router does not initiate inverse ATM ARP requests.

By default, inverse ATM ARP is disabled. To configure a VC to respond to inverse ATM ARP requests, include the inverse-arp statement at the [edit interfaces *interface-name* unit *logical-unit-number* family *family* address *address* multipoint-destination *destination-address*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family family address address
multipoint-destination destination-address]
inverse-arp;
```

Configure Inverse Frame Relay ARP

You can configure Frame Relay interfaces to support inverse Frame Relay ARP, as described in RFC 2390. When inverse Frame Relay ARP is enabled, the router responds to received inverse Frame Relay ARP requests by providing IP address information to the requesting Frame Relay device.

The router does not initiate inverse Frame Relay ARP requests.

By default, inverse Frame Relay ARP is disabled. To configure a VC to respond to inverse Frame Relay ARP requests, include the inverse-arp statement at the [edit interfaces *interface-name* unit *logical-unit-number* family *family* address *address* multipoint-destination *destination-address*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family family address address
multipoint-destination destination-address]
inverse-arp;
```

Define the ATM Traffic-Shaping Profile

When you use an ATM encapsulation, you can configure a traffic-shaping profile that defines the following:

- Bandwidth utilization, which consists of either a constant rate, or a peak cell rate with sustained cell rate and burst tolerance
- Maximum queue length

These values are used in the ATM generic cell-rate algorithm, which is a leaky bucket algorithm that defines the short-term burst rate for ATM cells, the maximum number of cells that can be included in a burst, and the long-term sustained ATM cell traffic rate. Each individual VC has its own independent shaping parameters.

By default, the bandwidth utilization is unlimited; that is, unspecified bit rate (UBR) is used. Also, by default, buffer usage by VCs is unregulated. To define limits to bandwidth utilization on a point-to-point interface or to limit buffer use, include the shaping statement. For point-to-point interfaces, include the shaping statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level

```
[edit interfaces interface-name unit logical-unit-number]
shaping {
    (cbr rate | vbr peak rate sustained rate burst length);
    queue-length number;
}
```

For virtual circuits that are part of a point-to-multipoint interface, include the shaping statement at the [edit interfaces *interface-name* unit *logical-unit-number* family *family* address *address* multipoint-destination *destination-address*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family family address address
multipoint-destination destination-address]
shaping {
    (cbr rate | vbr peak rate sustained rate burst length);
    queue-length number;
}
```

When defining the ATM traffic-shaping profile, you can do the following:

Configure CBR on page 136

Configure VBR on page 137

Specify Shaping Values on page 137

Configure CBR

For traffic that does not require the ability to periodically burst to a higher rate, you can configure a constant bit rate (CBR) by including the cbr statement at the [edit interfaces *interface-name* unit *logical-unit-number* shaping] or [edit interfaces *interface-name* unit *logical-unit-number* family *family* address *address* shaping] hierarchy level:

```
cbr rate;
```

For more information, see “Specify Shaping Values” on page 137.

Configure VBR

To define variable bandwidth rate (VBR) utilization, include the vbr statement at the [edit interfaces *interface-name* unit *logical-unit-number*] or [edit interfaces *interface-name* unit *logical-unit-number* family *family* address *address*] hierarchy level:

```
vbr peak rate sustained rate burst length;
```

You can define the following VBR traffic-shaping properties:

Peak rate—Top rate at which traffic can burst.

Sustained rate—Normal traffic rate averaged over time.

Burst length—Maximum number of cells that a burst of traffic can contain. It can be a value from 1 through 255 cells.

Specify Shaping Values

You can specify the rates in bits per second (bps) or cells per second (cps). For OC-3c interfaces, the highest rate is 135,600,000 bps (353,125 cps), which corresponds to 100 percent of the available line rate. For OC-12c interfaces, the highest rate is 271,263,396 bps (706,415.09 cps), which corresponds to 50 percent of the available line rate. Table 12 lists some of the other rates you can specify. If you specify a rate that is not listed, it is rounded to the nearest rate.

The exact number of values differs between OC-12c and OC-3c interfaces. OC-12c interfaces have about four times as many value increments as OC-3c interfaces. For OC-12c rates between 1/2 and 1/128 of the line rate, there are 128 steps between each 1/*n* value. For rates smaller than 1/128, there are (16,384 minus 128) or 16,256 values. This calculation is valid because fractional shaping is ignored at rates below 1/128. This results in about 32,384 distinct rates for OC-12c. For OC-3c, the starting point is full line rate, the fraction/integer breakpoint is about 1/32, and there is a maximum of 4096 scheduler slots, producing about 8032 distinct values.

In general, the actual packet rate on the interface is calculated with the following formula:

```
actual-rate = (128 * line-rate) / (trunc ((128 * line-rate) / desired-rate))
```

line-rate is the maximum available rate on the interface (in bits per second) after factoring out the overhead for SONET and ATM (per-cell) overheads. For OC-3c interfaces, the line rate is calculated as follows:

```
line-rate = 155,520,000 bps x (26/27) x (48/53) = 135,631,698.1 bps
```

For OC-12c interfaces, the line rate is calculated as follows:

```
line-rate = 622,080,000 bps x (26/27) x (48/53) = 542,526,792.45 bps
```

desired-rate is the rate you enter in the vbr statement, in bits per second.

The trunc operator indicates that all digits to the right of the decimal point should be dropped.

For OC-3c interfaces, the maximum available rate is 100 percent of *line-rate*, or 135,600,000 bps. For OC-12c interfaces, the maximum available rate is 50 percent of *line-rate*, or 271,263,396 bps.

The following example shows the calculations for determining the actual rate when the desired rate is 80 percent of the maximum rate:

OC-3c:

135,600,000 bps * 0.8 = 108,480,000 bps
actual-rate = (128 * 135,600,000.1) / (trunc ((128 * 135,600,000.1) / 108,480,000))
actual-rate = 17,356,800,013 / (trunc (17,356,800,013 / 108,480,000))
actual-rate = 17,356,800,013 / 160
actual-rate = 108,480,000 bps

OC-12c:

271,263,396 bps * 0.8 = 217,010,716.8 bps
actual-rate = (128 * 542,526,792.45) / (trunc ((128 * 542,526,792.45)/217,010,716.8))
actual-rate = 69,443,429,434 / (trunc (69,443,429,434 / 217,010,716.8))
actual-rate = 69,443,429,434 / 320
actual-rate = 217,010,717 bps

Table 12: Traffic-Shaping Rates

Interface Type	Line Rate (bps)	Line Rate (cps)	Percentage of Total Line Rate
OC-3			
	135,631,698	353,207.55	100.00
	134,580,290	350,469.50	99.22
	133,545,057	347,773.58	98.46
	132,525,629	345,118.82	97.71
	131,521,647	342,504.29	96.97
	130,532,762	339,929.07	96.24
	129,558,637	337,392.28	95.52
	128,598,943	334,893.08	94.81
	127,653,363	332,430.63	94.12
	126,721,587	330,004.13	93.43
OC-12			
	271,263,396	706,415.09	50.00
	270,207,897	703,666.40	49.81
	269,160,579	700,939.01	49.61
	268,121,349	698,232.68	49.42
	267,090,113	695,547.17	49.23
	266,066,779	692,882.24	49.04
	265,051,257	690,237.65	48.85
	264,043,458	687,613.17	48.67
	263,043,293	685,008.58	48.48
	262,050,677	682,423.64	48.30

Buffers are shared among all VCs, and by default, there is no limit to the buffer size for a VC. If a VC is particularly slow, it might use all the buffer resources. To limit the queue size of a particular VC, include the queue-length statement when configuring the VC at the [edit interfaces *interface-name* unit *logical-unit-number* shaping] or [edit interfaces *interface-name* unit *logical-unit-number* family *family* address *address* multipoint-destination *destination-address* shaping] hierarchy level:

```
queue-length number;
```

The length can range from 1 through 16,383 packets. The default is 16,383 packets.

Define the ATM OAM F5 Loopback Cell Period

When you are using an ATM encapsulation, you can configure the OAM F5 loopback cell period on virtual circuits, which is the interval at which OAM F5 loopback cells are transmitted.

By default, no OAM F5 loopback cells are sent. To send OAM F5 loopback cells on a point-to-point interface, include the oam-period statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]
oam-period (disable | seconds);
```

To send OAM F5 loopback cells on a virtual circuit that is part of a point-to-multipoint interface, include the oam-period statement at the [edit interfaces *interface-name* unit *logical-unit-number* family *family* address *address* multipoint-destination *destination-address*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family family address address
multipoint-destination destination-address]
oam-period (disable | seconds);
```

The period can range from 1 through 900 seconds. You can also enter oam-period disable, which disables the OAM loopback cell transmit feature.

OAM VC-AIS (alarm indication signal) and VC-RDI (remote defect indication) defect indication cells are used for identifying and reporting VC defects end-to-end. When a physical link or interface failure occurs, intermediate nodes insert OAM AIS cells into all the downstream VCs affected by the failure. Upon receiving an AIS cell on a VC, the router marks the logical interface down and sends an RDI cell on the same VC to let the remote end know the error status. When an RDI cell is received on a VC, the router sets the logical interface status to down. When no AIS or RDI cells are received for 3 seconds, the router sets the logical interface status to up. You do not need to configure anything to enable defect indication.

Configure the ATM OAM F5 Loopback Cell Threshold

When you are using an ATM encapsulation, you can configure the OAM F5 loopback cell threshold on VCs, which is the minimum number of consecutive OAM F5 loopback cells received before declaring that a VC is up or lost before declaring that a VC is down.

By default, when five consecutive OAM F5 loopback cells are received, the VC is considered to be up, and when five consecutive cells are lost, the VC is considered to be down. To modify these values on a point-to-point interface, include the oam-liveness statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]
oam-liveness {
    up-count cells;
    down-count cells;
}
```

To modify the OAM F5 loopback cell count threshold on a virtual circuit that is part of a point-to-multipoint interface, include the oam-liveness statement at the [edit interfaces *interface-name* unit *logical-unit-number* family *family* address *address* address *address* multipoint-destination *destination-address*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family family address address
multipoint-destination destination-address]
oam-liveness {
    up-count cells;
    down-count cells;
}
```

The cell count can be a value from 1 through 255 cells.

Configure a Point-to-Multipoint Frame Relay Connection

To configure a point-to-multipoint Frame Relay connection (also called a multipoint NBMA connection), include the multipoint-destination statement within the address statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]
address address {
    multipoint-destination destination-address dlci dlci-identifier;
}
```

address is the interface's address.

For each destination, include one multipoint-destination statement. *destination-address* is the address of the remote side of the connection, and *dlci-identifier* is the DLCI identifier for the connection.

When you are configuring point-to-multipoint connections, all interfaces in the subnet must use the same MTU size.

If keepalives are enabled, causing the interface to send LMI messages during idle times, the number of possible DLCI configurations is limited by the MTU selected for the interface. For more information, see "Configure Keepalives" on page 50.

Configure Static ARP Table Entries

For Ethernet, Fast Ethernet, and Gigabit Ethernet interfaces, you can configure static ARP table entries, defining mappings between IP and MAC addresses. To configure static ARP table entries, include the arp statement at the [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family inet address address]
arp ip-address (mac | multicast-mac) mac-address <publish>;
```

The IP address that you specify must be part of the subnet defined in the enclosing address statement.

To associate a multicast MAC address with a unicast IP address, include the multicast-mac statement.

Specify the MAC address as six hexadecimal bytes in one of the following formats: *nnnn.nnnn.nnnn* or *nn:nn:nn:nn:nn:nn*. For example, 0011.2233.4455 or 00:11:22:33:44:55.

For unicast MAC addresses only, if you include the publish option, the router replies to proxy ARP requests.

Example: Configure Static ARP Table Entries

Configure two static ARP table entries on the router's management interface:

```
interfaces fxp0 {
    unit 0 {
        family inet {
            address 10.10.0.11/24 {
                arp 10.10.0.99 mac 0001.0002.0003;
                arp 10.10.0.101 mac 00:11:22:33:44:55 publish;
            }
        }
    }
}
```

Apply Policers

To apply policers to an interface, include the policer statement when configuring the logical interface at the [edit interfaces *interface-name* unit *logical-unit-number* family inet] hierarchy level:

```
[edit interfaces]
interfaces interface-name {
    unit logical-unit-number {
        family inet {
            policer {
                input policer-template-name;
                output policer-template-name;
            }
        }
    }
}
```

To use policing on a CCC or TCC interface, you must include the family (ccc | tcc) statement at the [edit interfaces *interface-name* unit *logical-unit-number* family inet] hierarchy level.

For CCC and TCC interfaces, you can configure input policers only.

In the input statement, list the name of one policer template to be evaluated when packets are received on the interface.

In the output statement, list the name of one policer template to be evaluated when packets are transmitted on the interface.

You can configure a different policer on each protocol family under an interface. You can configure one input policer only and one output policer only for each protocol family. You can use the same policer one or more times.

If you apply both policers and firewall filters to an interface, policers are evaluated closest to the wire: Input policers are evaluated before input firewall filters, and output policers are evaluated after output firewall filters.

If you apply the policer to the interface lo0, it is applied to packets received or transmitted by the Routing Engine.

For more information about policers, see the *JUNOS Internet Software Configuration Guide: Policy Framework*.

Apply Firewall Filters

To apply firewall filters to an interface, include the filter statement when configuring the logical interface at the [edit interfaces *interface-name* unit *logical-unit-number* family inet] hierarchy level:

```
[edit interfaces]
interfaces interface-name {
    unit logical-unit-number {
        family inet {
            filter {
                group filter-group-number;
                input filter-name;
                output filter-name;
            }
        }
    }
}
```

In the group statement, specify the interface group number to associate with the filter.

In the input statement, list the name of one firewall filter to be evaluated when packets are received on the interface.

In the output statement, list the name of one firewall filter to be evaluated when packets are transmitted on the interface.

You can use the same filter one or more times.

If you apply the filter to the interface lo0, it is applied to packets received or transmitted by the Routing Engine.

For more information about firewall filters, see the *JUNOS Internet Software Configuration Guide: Policy Framework*.

Define Interface Groups in Firewall Filters

When applying a firewall filter, you can define an interface to be part of an *interface group*. Packets received on that interface are tagged as being part of the group. You can then match these packets using the interface-group match statement, as described in the *JUNOS Internet Software Configuration Guide: Policy Framework*.

To define the interface to be part of an interface group, include the group statement at the [edit interfaces *interface-name* unit *logical-unit-number* family inet filter] hierarchy level:

```
[edit interfaces]
interfaces interface-name {
    unit logical-unit-number {
        family inet {
            filter {
                group filter-group-number;
            }
        }
    }
}
```

Configure Ethernet TCC and Extended VLAN TCC

For Layer 2.5 VPNs employing an Ethernet interface as the TCC router, you can configure an Ethernet translational cross-connect (TCC) or an extended VLAN TCC.

To configure an Ethernet TCC, use the ethernet-tcc encapsulation type at the [edit interfaces interface-name] hierarchy level. To configure an extended VLAN TCC, use the extended-vlan-tcc encapsulation type at the [edit interfaces interface-name] hierarchy level.

To configure an Ethernet TCC or an extended VLAN TCC, include the remote statement at the [edit interfaces *interface-name* unit *logical-unit-number* family tcc] hierarchy level and the arp statement at the [edit interfaces *interface-name* unit *logical-unit-number* family *family* address *address*] hierarchy level:

```
[edit interfaces interfaces interface-name unit logical-unit-number family tcc]
remote {
    mac-address address;
}

[edit interfaces interface-name unit logical-unit-number family family address local-ip-address]
arp ip-address mac mac-address ;
```

The remote statement provides ARP capability from the TCC switching router to the Ethernet neighbor. mac-address is the the physical Layer 2 address of the Ethernet neighbor, also known as the remote router.

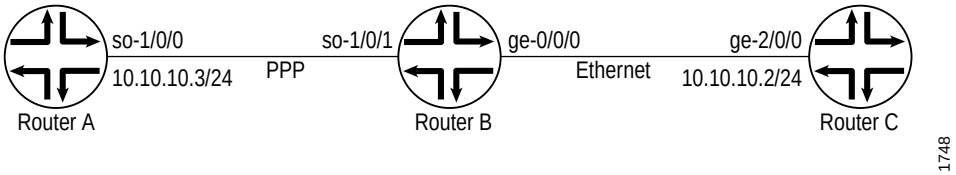
In the opposite direction, the arp statement on the Ethernet neighbor provides return path ARP functionality from the Ethernet neighbor to the TCC router. arp is the IP address of the non-Ethernet TCC neighbor, and mac is the MAC address of the TCC router's Ethernet interface.

Ethernet TCC is supported on interfaces that carry IPv4 traffic only. Ethernet TCC encapsulation is supported on one-port Gigabit Ethernet, two-port Gigabit Ethernet, four-port Gigabit Ethernet, and four-port Fast Ethernet PICs only. Ethernet TCC is not supported on the T-series platforms.

Example: Configure Ethernet TCC

Configure a full-duplex Layer 2.5 translational cross-connect between Router A and Router C, using a Juniper Networks router, Router B, as the TCC interface. Ethernet TCC encapsulation provides an Ethernet wide area circuit for interconnecting IP traffic. See the topology in Figure 8.

Figure 8: Example Topology of Layer 2.5 Translational Cross-Connect




```
interfaces ge-0/0/0 {
    encapsulation ethernet-tcc;
    unit 0 {
        family tcc {
            remote {
                mac-address 0001.0002.0003;
            }
        }
    }
}
```

The Router A–to–Router B circuit is PPP, and the Router B–to–Router C circuit accepts packets carrying standard Tag Protocol ID (TPID) values.

If traffic is in the direction from Router A to Router C, the JUNOS software strips all PPP encapsulation data from incoming packets and adds Ethernet encapsulation data before forwarding the packets. In the direction from Router C to Router A, the JUNOS software strips all Ethernet encapsulation data from incoming packets and adds PPP encapsulation data before forwarding the packets.

Configure Static ARP Table Entry for Router C

Ethernet TCC does not look at Layer 3 IP addresses; therefore, for the above example to work, you must configure a static ARP table entry, defining a mapping between the IP and MAC addresses of Router C, as shown in the following example:

```
interfaces ge-2/0/0 {
    unit 0 {
        family inet {
            address 10.10.10.2/24; {
                arp 10.10.10.3 mac 0011.2233.4455;
            }
        }
    }
}
```

0011.2233.4455 is the MAC address of Router B's ge-0/0/0 interface.

10.10.10.3 is the IP address of Router A's so-1/0/0 interface.

When Router C sends packets to 10.10.10.3 on Router A, the packet's first go to 0011.2233.4455 on Router B's. From Router B, the packets are forwarded to Router A.

For more information about static ARP, see "Configure Static ARP Table Entries" on page 141. For a configuration example showing an extended VLAN TCC, see "Example 2: Configure Extended VLAN TCC Encapsulation" on page 237. For more information about Layer 2.5 VPNs, see the *JUNOS Internet Softw are Configur ation Guide: VPNs* .

Configure Unicast Reverse Path Forwarding

For interfaces that carry IPv4 or IPv6 traffic, you can reduce the impact of denial-of-service (DoS) attacks by configuring unicast reverse path forwarding (RPF). Unicast RPF helps determine the source of attacks and rejects packets from unexpected source addresses on interfaces where unicast RPF is enabled

If you enable unicast RPF on live traffic, some packets are dropped while the Packet Forwarding Engine is updating.

To configure unicast RPF, your router must be equipped with the Internet Processor II ASIC.

To configure unicast RPF, include the rpf-check statement at the [edit interfaces *interface-name* unit *logical-unit-number* family inet] or [edit interfaces *interface-name* unit *logical-unit-number* family inet6] hierarchy level:

```
rpf-check fail-filter filter-name;
```

The fail-filter statement allows you to apply a filter to packets that fail the unicast RPF check. You can define the fail filter to perform any filter operation, including accept, reject, log, sample, or police.

Unicast RPF has several consequences when implemented with traffic filters:

RPF fail filters are evaluated after input filters and before output filters.

If you configure a filter counter for packets dropped by an input filter, and you want to know the total number of packets dropped, you must also configure a filter counter for packets dropped by the RPF check.

To count packets that fail the RPF check and are accepted by the RPF fail filter, you must configure a filter counter.

If an input filter forwards packets anywhere other than inet.0 or inet6.0, the unicast RPF check is not performed.

For more information about unicast RPF fail filters, see "Accept DHCP and BOOTP Packets" on page 146.

Accept DHCP and BOOTP Packets

To accept Bootstrap Protocol (BOOTP) packets and Dynamic Host Configuration Protocol (DHCP) packets, you must define a filter that accepts all packets with a source address of 0.0.0.0 and a destination address of 255.255.255.255, as shown in the following example:

```
[edit firewall]
filter rpf-special-case-dhcp-bootp {
    term allow-dhcp-bootp {
        from {
            source-address {
                0.0.0.0/32;
            }
            destination-address {
                255.255.255.255/32;
            }
        }
        then {
            count rpf-dhcp-bootp-traffic;
            accept;
        }
    }
    term default {
        then {
            log;
            reject;
        }
    }
}

[edit]
interfaces {
    so-0/0/0 {
        unit 0 {
            family inet {
                rpf-check fail-filter rpf-special-case-dhcp-bootp;
            }
        }
    }
}
```

Configure Multicast Tunnels

For interfaces that carry IPv4 or IPv6 traffic, you can configure multicast tunnels. To configure a multicast tunnel, include the multicasts-only statement at the [edit interfaces *interface-name* unit *logical-unit-number* family inet] or [edit interfaces *interface-name* unit *logical-unit-number* family inet6] hierarchy level:

```
multicasts-only;
```

Multicast tunnels filter all unicast packets; if an incoming packet is not destined for a 224/8 or greater prefix, the packet is dropped and a counter is incremented.

You can configure this property on GRE, IP-IP, PIM, and MT tunnels only.

Enable Source Class and Destination Class Usage

For interfaces that carry IPv4 traffic, you can maintain packet counts based on the entry and exit points for traffic passing through your network. Entry and exit points are identified by source and destination prefixes grouped into disjoint sets defined as *source classes* and *destination classes* . You can define classes based on a variety of parameters, such as routing neighbors, autonomous systems, and route filters.

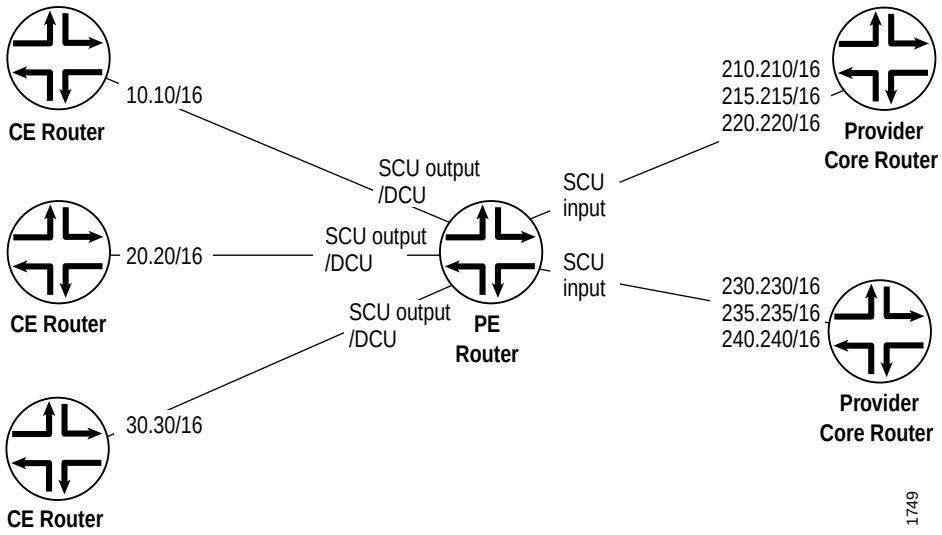
Source class usage (SCU) counts packets sent to customers by performing lookup on the IP source address and the IP destination address. SCU makes it possible to track traffic originating from specific prefixes on the provider core and destined for specific prefixes on the customer edge. You must enable SCU accounting on both the inbound and outbound physical interfaces.

Destination class usage (DCU) counts packets from customers by performing lookup of the IP destination address. DCU makes it possible to track traffic originating from the customer edge and destined for specific prefixes on the provider core router.

Figure 9 illustrates an ISP network. In this topology, you can use DCU to count packets customers send to specific prefixes. For example, you can have three counters, one per customer, that count the packets destined for prefix 210.210/16 and 220.220/16.

You can use SCU to count packets the provider sends from specific prefixes. For example, you can count the packets sent from prefix 210.210/16 and 215.215/16 and transmitted on a specific output interface.

Figure 9: Prefix Accounting with Source and Destination Classes



You can configure up to 126 source classes and 126 destination classes. For each interface on which you enable destination class usage and source class usage, the software maintains an interface-specific counter for each corresponding class up to the 126 class limit.

> **Note** To configure source class and destination class usage, your router must be equipped with the Internet Processor II ASIC.

To enable packet counting on an interface, include the accounting statement at the [edit interfaces *interface-name* unit *logical-unit-number* family inet] hierarchy level:

```
accounting {
    destination-class-usage;
    source-class-usage {
        (input | output | [input output]);
    }
}
```

For SCU to work, you must configure at least one input interface and at least one output interface. An incoming packet is counted only once, and SCU takes priority over DCU. This means that when a packet arrives on an interface on which you include the source-class-usage input and destination-class-usage statements in the configuration, and when the source and destination both match accounting prefixes, the software associates the packet with the source class only. To ensure the outgoing packet is counted, include the source-class-usage output statements in the configuration of the outgoing interface.

Once you enable accounting on an interface, the software maintains packet counters for that interface. You must then configure the source class and destination class attributes in policy action statements, which must be included in forwarding-table export policies. For a complete discussion about source and destination class accounting profiles, see the *JUNOS Internet Software Configuration Guide: Network Management*.

Example 1: Enable Source Class Usage and Destination Class Usage

Configure DCU and SCU output on one interface:

```
[edit]
interfaces {
    so-6/1/0 {
        unit 0 {
            family inet {
                accounting {
                    destination-class-usage;
                    source-class-usage {
                        output;
                    }
                }
            }
        }
    }
}
```

Configure SCU input on another interface:

```
[edit]
interfaces {
    ge-0/1/0 {
        unit 0 {
            family inet {
                accounting {
                    source-class-usage {
                        input;
                    }
                }
            }
        }
    }
}
```

Optionally, you can include the input and output statements on a single interface:

```
[edit]
interfaces {
    ge-0/1/2 {
        unit 0 {
            family inet {
                accounting {
                    source-class-usage {
                        input;
                        output;
                    }
                }
            }
        }
    }
}
```

Example 2: Enable Packet Counting for Layer 3 VPNs

You can use SCU and DCU to count packets on Layer 3 VPNs. To enable packet counting for Layer 3 VPN implementations at the egress point of the MPLS tunnel, you must configure a VPN loopback tunnel interface (vt) on the PE router, map the VRF instance type to the VPN loopback tunnel interface, and send the traffic received from the VPN out the source class output interface, as shown in the following example:

1. Configure a VPN loopback tunnel interface on a provider edge router equipped with a tunnel PIC:

```
[edit interfaces]
vt-0/3/0 {
    unit 0 {
        family inet {
            accounting {
                source-class-usage {
                    input;
                }
            }
        }
    }
}
```

2. Map the VRF instance type to the VPN loopback tunnel interface:

```
[edit]
routing-instances {
    routing-instance-name {
        instance-type vrf;
        interface at-2/1/1.0;
        interface vt-0/3/0.0;
        route-distinguisher 10.255.14.225:100;
        vrf-import import-policy-name;
        vrf-export export-policy-name;
        protocols {
            bgp {
                group to-r4 {
                    local-address 10.27.253.1;
                    peer-as 400;
                    neighbor 10.27.253.2;
                }
            }
        }
    }
}
```

For SCU and DCU to work, you must not include the vrf-table-label statement at the [edit routing-instances *instance-name*] hierarchy level.

3. Send traffic received from the VPN out the source class output interface:

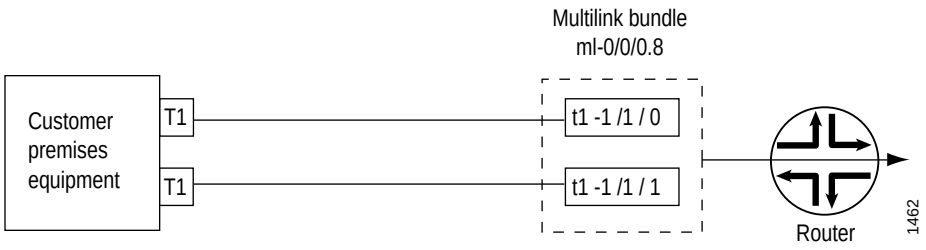
```
[edit interfaces]
at-1/1/0 {
    unit 0 {
        family inet {
            accounting {
                source-class-usage {
                    output;
                }
            }
        }
    }
}
```

For more information about VPNs, see the *JUNOS Internet Softw are Configur ation Guide: VPNs*. For more information about VPN loopback tunnel interfaces, see "Configure Tunnel Interfaces" on page 311.

Configure Multilink Interfaces

To complete a multilink configuration, you need to configure both the physical link, such as a T1 or E1 connection, and the multilink bundle, which is a logical connection, as shown in Figure 10. The physical link is usually connected to networks capable of supporting the Multilink Point-to-Point Protocol (MLPPP) or Multilink Frame Relay (MLFR).

Figure 10: Multilink Interface Configuration



Using the topology in Figure 10 as an example, configure a multilink interface over a T1 connection as follows:

1. To configure a physical T1 link for MLPPP, include the following statements at the [edit interfaces] hierarchy level:

   ```
   [edit interfaces t1-fpc/pic/port]
   unit 0 {
      family mlppp {
         bundle ml-fpc/pic/port;
      }
   }
   ```

   You do not need to set any IP address on this link.

   To configure a physical T1 link for MLFR, include the following statements at the [edit interfaces] hierarchy level:

   ```
   [edit interfaces t1-fpc/pic/port]
   unit 0 {
      dlci dlci-identifier;
      encapsulation multilink-framerelay;
      family mlfr {
         bundle ml-fpc/pic/port;
      }
   }
   ```

   You do not need to set any IP address on this link.

2. To configure the logical address for the MLPPP bundle, include the following statements at the [edit interfaces ml-*fpc/pic/port* unit *logical-unit-number*] hierarchy level:

   ```
   [edit interfaces ml-fpc/pic/port unit logical-unit-number]
   family inet {
      address address {
         destination address;
      }
   }
   ```

   For MLPPP, when certain options such as MRRU arrive on the T1 interface, this configuration initiates a bundle-join. After a few sanity checks, the T1 interface effectively becomes a part of the multilink bundle.

   To configure the logical address for the MLFR bundle, include the following statements at the [edit interfaces ml-*fpc/pic/port* unit *logical-unit-number*] hierarchy level:

   ```
   [edit interfaces ml-fpc/pic/port unit logical-unit-number]
   encapsulation multilink-framerelay;
   family inet {
      address address {
         destination address;
      }
   }
   ```

For MLPPP and MLFR links, you must specify the address as /32 or /30. Other subnet designations are treated as a mismatch.

To configure encapsulation and other multilink properties that are specified at the [edit interfaces ml-*fpc/pic/port* unit *logical-unit-number*] hierarchy level, see "Configure Logical Interface Properties" on page 99.

Configure Security Associations

To use IPSec security services, you create a security association (SA) between hosts. An SA is a simplex connection that allows two hosts to communicate with each other securely by means of IPSec. You can configure two types of SAs:

Manual—Requires no negotiation; all values, including the keys, are static and specified in the configuration. As a result, each peer must have the same configured options for communication to take place.

Dynamic—Specifies proposals to be negotiated with the tunnel peer. The keys are generated as part of the negotiation and therefore do not need to be specified in the configuration. The dynamic SA includes one or more proposal statements, which allow you to prioritize a list of protocols and algorithms to be negotiated with the peer.

We recommend that you configure no more than 512 dynamic security associations per ES PIC.

To configure an SA for IPSec, include the security-association statement and specify a security association name at the [edit security ipsec] hierarchy level:

```
[edit security ipsec]
security-association name;
```

To configure encryption interfaces, you associate the security profile with the interface by including the ipsec-sa *sa-name* statement at the [edit interfaces es-*fpc/pic/port* unit *logical-unit-number* family inet] hierarchy level:

```
[edit interfaces es-fpc/pic/port unit logical-unit-number family inet]
ipsec-sa sa-name;
```

For more information about the ES PIC, see "Configure Encryption Interfaces" on page 223. For detailed information on Internet Protocol Security (IPSec) and configuring the security association, see the *JUNOS Internet Softw are Configur ation Guide: Getting St arted*.

Configure VRRP

For Ethernet, Fast Ethernet, and Gigabit Ethernet interfaces, you can configure the Virtual Router Redundancy Protocol (VRRP). VRRP allows hosts on a LAN to make use of redundant routers on that LAN without requiring more than the static configuration of a single default route on the hosts. The VRRP routers share the IP address corresponding to the default route configured on the hosts. At any time, one of the VRRP routers is the master (active) and the others are backups. If the master fails, one of the backup routers becomes the new master, thus always providing a virtual default router and allowing traffic on the LAN to be routed without relying on a single router.

VRRP is defined in the following document:

RFC 2338, *Virtual Router Redundanc y Protocol*

To configure VRRP, include the vrrp-group statement at the [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family inet address address]
vrrp-group group-number {
    virtual-address [ addresses ];
    priority number;
    (accept-data | no-accept-data);
    advertise-interval seconds;
    authentication-type authentication;
    authentication-key key;
    (preempt | no-preempt);
    track {
        interface interface-name priority-cost cost;
    }
}
```

To trace VRRP operations, include the traceoptions statement at the [edit protocols vrrp] hierarchy level:

```
[edit protocols vrrp traceoptions]
    file {
        filename filename;
        files number;
        size size;
        (world-readable | no-world-readable);
    }
    flag flag;
```

For more information, see “Trace VRRP Operations” on page 159.

You can configure the following VRRP properties:

Configure Basic VRRP Support on page 155

Configure VRRP Authentication on page 156

Configure the Advertisement Interval for the VRRP Master Router on page 157

Configure a Backup Router to Preempt the Master Router on page 157

Accept Packets Destined for the Virtual IP Address on page 157

Configure an Interface to Be Tracked on page 158

Trace VRRP Operations on page 159

For a VRRP configuration example, see “Example: Configure VRRP” on page 160.

Configure Basic VRRP Support

To configure basic VRRP support, configure VRRP groups on interfaces by including the following statements at the [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family inet address address]
vrrp-group group-number {
    virtual-address [ addresses ];
    priority number;
}
```

An interface can be a member of one or more VRRP groups. For each group, you must configure the following:

Group number—Identifies the VRRP group. It can be a value from 0 through 255.

If you also enable MAC source address filtering on the interface, as described in “Configure MAC Address Filtering” on page 79, you must include the virtual MAC address in the list of source MAC addresses that you specify in the source-address-filter statement. MAC addresses ranging from 00:00:5e:00:01:00 through 00:00:5e:00:01:ff are reserved for VRRP, as defined in RFC 2338. The VRRP group number must be the decimal equivalent of the last hexadecimal byte of the virtual MAC address.

Addresses of one or more virtual routers that are members of the VRRP group—Virtual IP addresses associated with the virtual router in the VRRP group. Normally, you configure only one virtual IP address per group. The virtual IP addresses must be the same for all routers in the VRRP group.

In the addresses, specify the address only. Do not include a prefix length.

If you configure a virtual IP address to be the same as the interface’s address (the address configured with the address statement), the interface becomes the master virtual router for the group. In this case, you must configure the priority to be 255 and you must configure preemption by including the preempt statement. If you have multiple VRRP groups on an interface, the interface can be the master virtual router for only one of the groups.

If the virtual IP address you choose is not the same as the interface's address, you must ensure that this address does not appear anywhere else in the router's configuration. Check that you do not use this address for other interfaces, for the IP address of a tunnel, or for the IP address of static ARP entries.

Priority for this router to become the master virtual router—Value used to elect the master virtual router in the VRRP group. It can be a number from 1 through 255. The default value for backup routers is 100. A larger value indicates a higher priority. The router with the highest priority within the group becomes the master router.

Within a single VRRP group, the master and backup routers cannot be the same router.

Configure VRRP Authentication

All VRRP protocol exchanges can be authenticated to guarantee that only trusted routers participate in the AS's routing. By default, VRRP authentication is disabled. You can configure one of the following authentication methods; each VRRP group must use the same method:

Simple authentication—Uses a text password included in the transmitted packet. The receiving router uses an authentication key (password) to verify the packet.

MD5 algorithm—Creates the authentication data field in the IP authentication header. This header is used to encapsulate the VRRP protocol data unit (PDU). The receiving router uses an authentication key (password) to verify the authenticity of the IP authentication header and VRRP PDU.

To enable authentication and specify an authentication method, include the authentication-type statement at the [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family inet address address
vrrp-group group-number]
authentication-type authentication;
```

authentication can be none, simple, or md5. The authentication type must be the same for all routers in the VRRP group.

If you included the authentication-type statement to select an authentication method, you can configure a key (password) on each interface by including the authentication-key statement at the [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family inet address address
vrrp-group group-number]
authentication-key key;
```

The key (password) is an ASCII string. For simple authentication, it can be 1 through 8 characters long. For MD-5 authentication, it can be 1 through 16 characters long. If you include spaces, enclose all characters in quotation marks (" "). The key must be the same for all routers in the VRRP group.

Configure the Advertisement Interval for the VRRP Master Router

By default, the master router sends VRRP advertisement packets every second to all members of the VRRP group. These packets indicate that the master router is still operational. If the master router fails or becomes unreachable, the backup router with the highest priority value becomes the new master router.

To modify the time between the sending of VRRP advertisement packets, include the advertise-interval statement at the [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family inet address address
vrrp-group group-number]
advertise-interval seconds;
```

The interval can range from 1 through 255 seconds. The interval must be the same for all routers in the VRRP group.

Configure a Backup Router to Preempt the Master Router

By default, a higher priority backup router preempts a lower priority master router. To explicitly allow the master router to be preempted, include the preempt statement at the [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family inet address address
vrrp-group group-number]
preempt;
```

To prohibit a higher priority backup router from preempting a lower priority master router, include the no-preempt statement at the [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family inet address address
vrrp-group group-number]
no-preempt;
```

The router that owns the IP address(es) associated with the virtual router always preempts, independent of the setting of the (preempt | no-preempt) flag.

Accept Packets Destined for the Virtual IP Address

To configure an interface to accept packets destined for the virtual IP address, include the accept-data statement at the [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family inet address address
vrrp-group group-number]
accept-data;
```

To prohibit the interface from accepting packets destined for the virtual IP address, include the no-accept-data statement at the [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family inet address address
vrrp-group group-number]
no-accept-data;
```

The accept-data statement has the following consequences:

You do not need to include the accept-data statement to activate this feature if the master router owns the virtual IP address.

If you do not include the accept-data statement, and if the master router owns the virtual IP address, the master router responds to ICMP message requests only.

You cannot include the accept-data statement when the priority of the master router is set to 255.

To restrict incoming IP packets to ICMP only, you must configure firewall filters to accept only ICMP packets.

If you include the accept-data statement, your router configuration will not comply with RFC 2338.

If you include the accept-data statement, VRRP clients should be able to process Gratuitous ARP.

If you include the accept-data statement, VRRP clients should not use packets other than ARP replies to update their ARP cache.

Configure an Interface to Be Tracked

VRRP can track whether an interface is up or down and dynamically change the priority of the VRRP group based on the state of the tracked interface, which might trigger a new master router election.

When interface tracking is enabled, you cannot configure a priority of 255, thereby designating the master router. For each VRRP group, 1 through 10 interfaces can be tracked.

To configure an interface to be tracked, include the track statement at the [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family inet address address vrrp-group
group-number]
track {
    interface interface-name priority-cost cost;
}
```

The priority cost is the value to be subtracted from the configured VRRP priority when the tracked interface is down, forcing a new master router election. The cost can range from 1 through 254. The sum of the costs for all tracked interfaces or routes must be less than or equal to the configured priority of the VRRP group.

Trace VRRP Operations

To trace VRRP operations, include the traceoptions statement at the [edit protocols vrrp] hierarchy level.

By default, VRRP logs the error, dcd configuration, and routing socket events in a file in the /var/log directory. By default, this file is named /var/log/vrrpd. The default file size is 1MB, and three files are created before the first one gets overwritten.

To change the configuration of the logging file, include the file statement at the [edit protocols vrrp traceoptions] hierarchy level:

```
[edit protocols vrrp traceoptions]
   file {
      filename filename;
      files number;
      size size;
      (world-readable | no-world-readable);
   }
   flag flag;
```

You can specify the following VRRP tracing flags:

all—Trace all VRRP operations.

database—Trace all database changes.

general—Trace all general events.

interfaces—Trace all interface changes.

normal—Trace all normal events.

packets—Trace all packets sent and received.

state—Trace all state transitions.

timer—Trace all timer events.

Example: Configure VRRP

Configure one master (Router A) and one backup (Router B) router. Note that the address configured in the virtual-address statements differs from the addresses configured in the address statements.

On Router A:

```
[edit]
interfaces {
    ge-0/0/0 {
        unit 0 {
            family inet {
                address 192.168.1.20/24 {
                    vrrp-group 27 {
                        virtual-address 192.168.1.15;
                        priority 254;
                        authentication-type simple;
                        authentication-key booJUM;
                    }
                }
            }
        }
    }
}
```

On Router B:

```
[edit]
interfaces {
    ge-4/2/0 {
        unit 0 {
            family inet {
                address 192.168.1.24/24 {
                    vrrp-group 27 {
                        virtual-address 192.168.1.15;
                        priority 200;
                        authentication-type simple;
                        authentication-key booJUM;
                    }
                }
            }
        }
    }
}
```

When configuring multiple VRRP groups on an interface, configure one to be the master virtual router for that group:

```
[edit]
interfaces {
    ge-0/0/0 {
        unit 0 {
            family inet {
                address 192.168.1.20/24 {
                    vrrp-group 2 {
                        virtual-address 192.168.1.20;
                        priority 255;
                        advertise-interval 3;
                        preempt;
                    }
                    vrrp-group 10 {
                        virtual-address 192.168.1.55;
                        priority 201;
                        advertise-interval 3;
                    }
                    vrrp-group 1 {
                        virtual-address 192.168.1.54;
                        priority 22;
                        advertise-interval 4;
                    }
                }
            }
        }
    }
}
```

Configure VRRP and MAC source address filtering on a Gigabit Ethernet interface. The VRRP group number is the decimal equivalent of the last byte of the virtual MAC address.

```
[edit interfaces]
ge-5/2/0 {
    gigether-options {
        source-filtering;
        source-address-filter {
            00:00:5e:00:01:0a;<— Virtual MAC address
        }
    }
    unit 0 {
        family inet {
            address 192.168.1.10/24 {
                vrrp-group 10 { <— VRRP group number
                    virtual-address 192.168.1.10;
                    priority 255;
                    preempt;
                }
            }
        }
    }
}
```